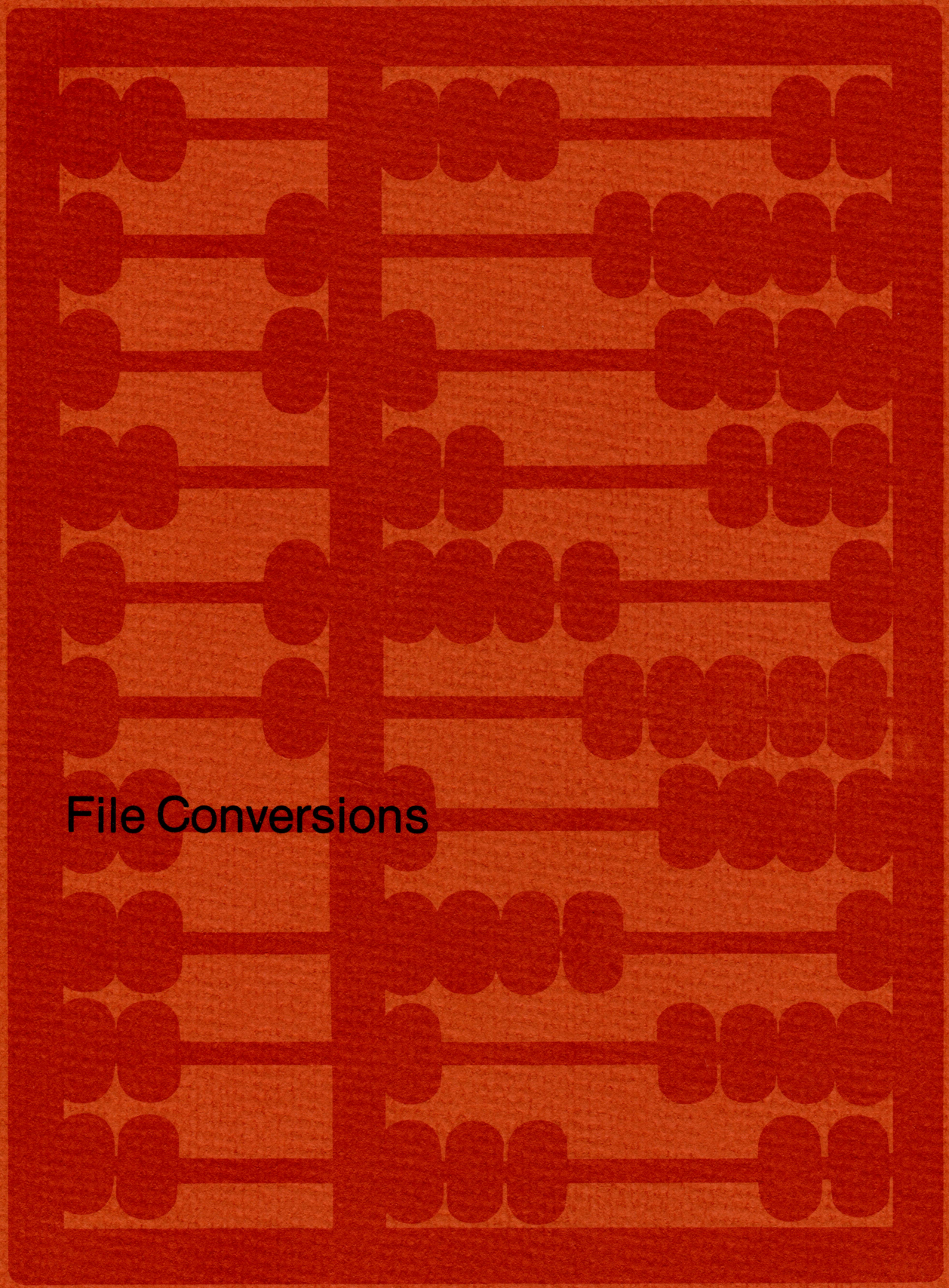


File Conversions

I.P. Sharp Associates Limited

FILE MOVES

EXTERNAL TO APL/APL TO EXTERNAL FILE CONVERSIONS

TABLE OF CONTENTS

Introduction	1
Why Conversion is Necessary	2
Four Common Types of Conversion	3
What Information You Must Supply	5
Further Notes on Conversion	9
Non-standard Conversions/APL Post Processing	10

Printed in Canada
Publication Code: 311FM01

INTRODUCTION

A file is a quantity of data stored on some permanent storage medium such as disk, magnetic tape or cards. The process of copying a file from one storage medium to another or converting an external file to a *SHARP APL* file (or vice versa) is called a **file move**.

All file moves are carried out by the I.P. Sharp operations staff. If you have an external file which you want converted to a *SHARP APL* data file or a *SHARP APL* file which you want converted to an external file, it is necessary to put in a request for a file move and provide the Sharp operator with enough information to carry out your request.

Several types of file moves are carried out by standard Sharp utility programs. For example, conversion from a simple sequential data file to a *SHARP APL* data file can usually be carried out using one of these utilities. Conversely, *SHARP APL* files can be converted to simple sequential files for use on our own or other systems. It is not necessary, however, to convert files from other systems running *SHARP APL*.

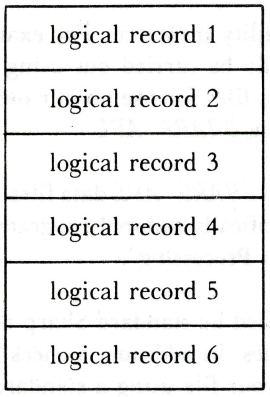
External files other than simple sequential can also be converted to *SHARP APL* data files. However, this process tends to be more complicated and may require the creation of a specifically tailored program or post processing in APL. See the section entitled 'Non-standard Conversions/ APL Post Processing'.

This manual discusses the types of conversion which can be handled by standard Sharp utilities and the actions to be taken if your conversion does not fall into one of these categories. In addition, a checklist is provided to help you in obtaining the information the operator will need to convert your file using a standard Sharp utility.

WHY CONVERSION IS NECESSARY

A *SHARP APL* file is made up of components, each of which can contain any APL object. For example, a component may hold an array of either character, integer, floating point, or Boolean data.

External sequential files, created, for example, by COBOL or FORTRAN programs, are structured somewhat differently. The structure of these files is described in terms of physical and logical records rather than components. For files stored on cards, one physical record is one logical record. With files stored on magnetic tape, however, a physical record is called a **block** and may contain more than one logical record as shown in the diagram below.



This is a physical record (block) containing fixed length logical records.

A simple sequential file is one in which the records consist uniformly of elements of those types represented in APL (Boolean, 32-bit integer, 64-bit double precision floating point, character). However, an external record may contain more than one data type, including some such as packed decimal, single precision floating point or 16-bit binary which have no corresponding type in *SHARP APL*.

FOUR COMMON TYPES OF CONVERSION

When converting an external file to a *SHARP APL* data file, a distinction should be made between files which have a fixed block length and those that don't. Files whose blocks vary in length are referred to as **unblocked** while files having a fixed or uniform blocksize are called **fixed block**. The way in which your file can be restructured is dependent on whether it is fixed block or unblocked. Conversely, when converting an APL file to an external file, you may have the choice of creating an unblocked or fixed block sequential file.

The options and restrictions involved in the conversion of an external file (fixed block or unblocked) to a *SHARP APL* file and vice versa are discussed below.

1. **Converting an External File (Fixed Block) to a SHARP APL File**

   If your file has a fixed block length (and the data type is uniform throughout the file) you have several choices as to what shape your APL components can be.

   a. A vector can be created by converting one physical record block to one APL component. The file is treated as if it were **unblocked**.

   b. A vector can be created by converting one logical record to one APL component. The number of elements per component must be specified. This is called **deblocking**.

   c. Matrices can be created by converting a specified number of logical records to the rows of a matrix. Two values have to be specified: the number of rows in the matrix and the number of elements in each row. This is called **reblocking**.

   The component data type can be specified as either character, integer, floating point, or Boolean. Note, however, that the data type of the external file must be homogeneous. If your external file contains, for example, both characters and integers, you must indicate whether you want your data to be treated as character or integer for the purpose of conversion. Once the conversion is complete, your file will require APL post processing to restore both data types.

   If the data type of your file is uniformly character, you have the option of translation from EBCDIC to Z CODE, APL's internal character set, (see 'Further Notes on Conversion'). It is also possible to request a special translate in which case you must provide your own translate table and its file name and component number.

2. **Converting an External File (Unblocked) to a SHARP APL File**

With an unblocked external file (i.e. containing variable length blocks), fewer options are available. Vectors of characters or integers are created by converting one block into one APL component. Any data representation is possible in the blocks of the external file, but the conversion treats all blocks as if they were the specified type. With character data, you have the option of translation from EBCDIC to Z CODE but the use of a special translate table is not allowed.

Raw M-entries (APL components complete with header information describing the component) can also be handled. One block (containing one M-entry) is converted to one APL component. Translation of character data is not possible. Note that this is a highly specialized format and unlikely to be applicable to your file conversion unless you are converting from an APL system which uses components similar to those of *SHARP APL*.

3. **Converting a SHARP APL File to an External File (Fixed Block)**

Components containing either fixed length vectors or matrices with a constant number of columns are converted to either logical records or blocks to create a fixed block file. The logical record size and blocksize must be specified. In the case of character data, if the logical record size differs from the component size, truncating or padding will take place.

The following options are available:

a. **Data Type** - this can be character, integer, floating point or Boolean. If the data type is not specified, the data type of the first component is used. Only components of the specified type will be converted. All other types will be ignored.

b. **Translation** - the three translate options (translate, no translate, and special translate) are available.

c. **Labels** - if your file is being written to magnetic tape, you can specify standard DOS labels, standard OS labels, unlabeled with no leading tape mark or unlabeled with leading tape mark.

d. **Recording Density** - if your file is being written to magnetic tape, you can specify 800 bpi, 1600 bpi or 6250 bpi. 1600 bpi will be used if no specification is given.

4. **Converting a SHARP APL File to an External File (Unblocked)**

In this type of conversion one APL component becomes one block. Each APL component consists of an array as well as shape and data type information describing that array. You have the choice of specifiying whether or not you wish this shape and data type information to be included on your external file. If you ask for arrays, it will be excluded, if you ask for M-entries it will be included.

M-entry blocks are always padded to at least 18 bytes. With array blocks, however, the conversion of a short data component (i.e. one under 18 bytes) is possible. This may result in the loss of a block when the tape is read. Therefore, you must make sure that such a component is padded to 18 bytes.

If your APL file contains an empty array (which is not being output as an M-entry) you must specify whether it will be converted to a tape mark or used to signal the end of conversion.

If your APL file contains characters you should specify whether or not they are to be translated. The four label options and the three recording densities listed in Section 3 above are available for files being written to magnetic tape.

WHAT INFORMATION YOU MUST SUPPLY

If your conversion falls into one of the four categories listed above, it can probably be handled by one of the Sharp utilities. To obtain all the information which the operator will need to carry out your conversion, go through the following checklist. Send the information to the operator with *)OPR, 666 BOX* or by contacting your local Sharp representative. Note that in the following checklist, a single asterisk beside an item indicates that this item applies to unblocked files only. A double asterisk indicates that the item applies to fixed block files only.

CHECKLIST

I External to SHARP APL File Conversion

1. Give the Volume Serial number of the external file.

2. Give the logical record size of the external file in bytes.

3. Give the blocksize of the external file in bytes.

4. If the external file is on a magnetic tape, what labels does it have?

   standard labels
   unlabeled

5. If the external file is on tape, give the density.

   800 bpi 1600 bpi 6250 bpi

6. What data types are stored on the external file?

   character
   integer
   floating point
   ** Boolean - must be character '0' or '1'
   M-entries

7. Choose an APL component shape:

   Vector - one block becomes one component.

   ** Vector - one logical record becomes one component. Specify the number of elements per component.

   ** Matrices - one logical record becomes one row of matrix. Specify the number of rows and the number of elements per row.

   * M-entries - one block becomes one component.

8. What APL data type will be used?

   character
   integer
   ** floating point
   ** Boolean

9. Do you require character translation? Specify:

translate
no translate
** special translate - give file name and component number

10. Provide account number and file name for the *SHARP APL* file. If the file exists, user number 1000 must have at least append access.

11. Provide the name, address and telephone number of the file owner or person who can answer technical questions concerning file format. This person will be notified when the conversion is complete.

NOTE: If you are unable to supply technical data such as data type, logical record length, etc., supplying a printout of the contents of the first few records of your file will help us to come up with this information.

II SHARP APL to External File Conversion

1. Provide the account number and the file name. Make sure user number 1000 has read access.

2. If the data is not being converted as M-entries, what data type is to be used?

   character
   integer
   floating point
   Boolean

   With fixed block files, if not specified, data type of the first component is used.

3. Do you require character translation? Specify:

   translate
   no translate (forced for M-entries)
   ** special translate - give file name and component number

4. How should the APL component be restructured?

   one vector component becomes one block
   ** one vector component becomes one logical record
   ** each row of matrix becomes one logical record
   * each M-entry component becomes one block

*5. When output blocks are not M-entries, empty component on *SHARP APL* file:

   becomes tape mark
   signals end of conversion

6. For fixed block output onto tape, specify the logical record size of the external file in bytes.

7. For fixed block output onto tape, specify the blocksize of the external file in bytes.

8. If output file is on tape, specify labels:

   standard DOS labels
   standard OS labels
   unlabeled with leading tape mark
   unlabeled without leading tape mark

9. If output file is on tape, specify density:

   800 bpi 1600 bpi 6250 bpi

10. Will you provide the tape? Borrow one of ours? Buy one of ours?

11. Provide the name, address and telephone number of the file owner or person who can answer technical questions concerning APL file format. This person will be notified when the conversion is complete.

FURTHER NOTES ON CONVERSION

1. The following table is used in character conversion from EBCDIC to Z CODE and vice versa:

EBCDIC	Z-CODE	EBCDIC	Z-CODE	EBCDIC	Z-CODE
A-Z	A-Z	a-z	A-Z	0-9	0-9
$	$	¢	¢	(11-0)	⊖
(12-0)	≠	(	(	)	)
;	;	/	/	"	"
+	+	-	-	%	÷
*	*	|	|	<	<
=	=	>	>	#	≠
@	α	&	∈	,	,
!	!	¬	~	—	—
.	.	:	:	'	'

Note 1: (11-0) refers to zero overpunched with minus. EBCDIC code x'D0'
(12-0) refers to zero overpunched with plus. EBCDIC code x'C0'

Note 2: When converting from APL to external (only)

¯ (high minus)	becomes - (minus)
⍲	becomes x'C0' (plus overpunched zero)
⌽	becomes ¢ (cent sign)

Note 3: When converting to APL, any characters which do not appear in table become ∇.
When converting from APL, any characters which do not appear in table are replaced by binary zeros.

2. Integer data are 4 byte (i.e. 32-bit) signed binary elements.

3. Fixed block records of character '0' and '1' may be converted to Boolean vectors or matrices. Internally, 32 Boolean values are stored (left justified) in a 32-bit word (4 bytes).

4. Floating point data are eight byte (64-bit) double precision elements.

NON-STANDARD CONVERSIONS/APL POST PROCESSING

If your file is very large or will be processed frequently or contains a complicated mixture of data types, it may be advantageous to have a special program written (usually COBOL) for the purpose of conversion. If the file contains many data types, most of the expense will be in getting the specifications complete and correct. Sharp programmers can provide you with any assistance you may need. Contact your local Sharp representative for further information.

With some conversions it is easier to convert the blocks to APL components in a straightforward way and then make any necessary adjustments to the file with APL post processing. For a one-time, urgently required conversion of a file which is not very large and which contains numeric values in character form, it may be advantageous to convert to an APL file of character vectors or matrices and then use built-in APL facilities $\square FI/\square VI$ to extract numeric values.

If your file is not too large but contains complex data types (i.e. packed decimals, single precision floating point, etc.) and conversion is urgently required, it may be necessary to have the data put into APL as untranslated vectors or matrices and then have interactive APL functions written to analyze the data and do the required conversions.

Your local *SHARP APL* representative can provide you with further information and advice concerning non-standard conversions and APL post processing. If will be helpful if you have available as much information on your file as possible.